BANCO DE DADOS RELACIONAL

OVERVIEW

LABORATÓRIO DE BANCO DE DADOS

VANESSA BORGES

Definição de banco de dados DATABASE

- **Definição: especifica um novo banco de dados**

  CREATE DATABASE <nome> [...];

  Normalmente, o criador se torna o dono do novo banco de dados

- **Alteração de um banco de dados existente**

  ALTER DATABASE name [[WITH] option [...]];

- **Remoção: remove um banco de dados existente**

  DROP DATABASE <nome> [option [...]];

https://www.postgresql.org/docs/current/sql-createdatabase.html

Definição de banco de dados SCHEMA

O PostgreSQL cria o schema PUBLIC como default

- **Definição: especifica um novo banco de dados**

  CREATE SCHEMA <nome>;

- **Alteração de um banco de dados existente**

  ALTER SCHEMA name [[WITH] option [...]];

- **Remoção: remove um banco de dados existente**

  DROP DATABASE <nome> [option [...]];

Banco de Dados Relacional

- Armazena e gerencia dados estruturados em tabelas bidimensionais.

- Essas tabelas são compostas por linhas (tuplas) e colunas (atributos), e possuem um esquema predefinido que determina o formato dos dados.

Exemplo: Tabela Cliente

clienteid	nome	email	cpf	telefone
1	João Silva	joao.silva@email.com	11111111111	123456789
2	Maria Oliveira	maria.oliveira@email.com	22222222222	987654321
3	Fernando	fernando@email.com	33333333333	987654321

Tupla

Atributos

Definição de tabela TABLE

- **Definir uma nova tabela**

```
CREATE [TEMPORARY | TEMP] TABLE <nome da tabela>
( [...,
<atributo> <tipo> [NULL|NOT NULL]
[restrição do atributo],
[CONSTRAINT <nome da restrição de tabela> <tipo da restrição
de tabela>, ...]);
```

- **Remover uma tabela existente**

```
DROP TABLE <nome da tabela>;
```

Exemplo:

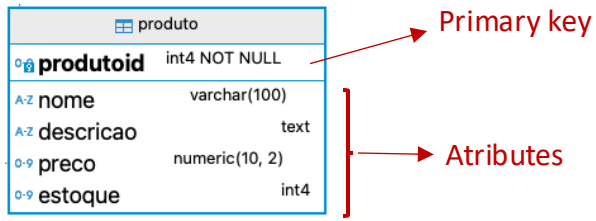
```
CREATE TABLE cliente (
        <declaração dos atributos, tipos e restrições>
nome        VARCHAR(100),
cpf         CHAR(11),
... ,
        <declaração das restrições de tabela>
PRIMARY KEY (clienteid),
UNIQUE (cpf),
...) ;
```

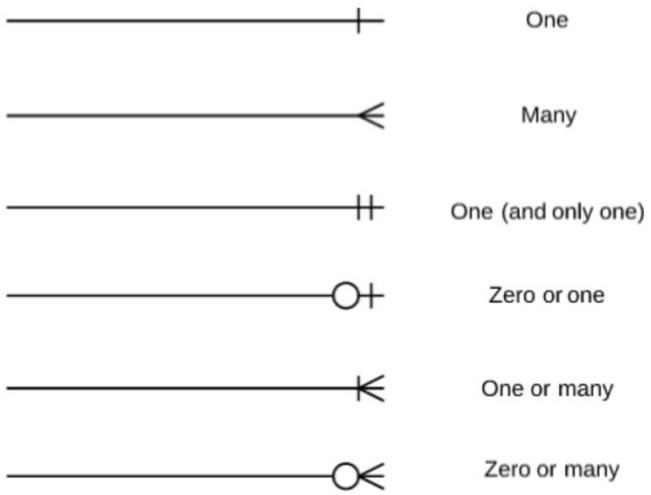
Definição de banco de dados DATABASE

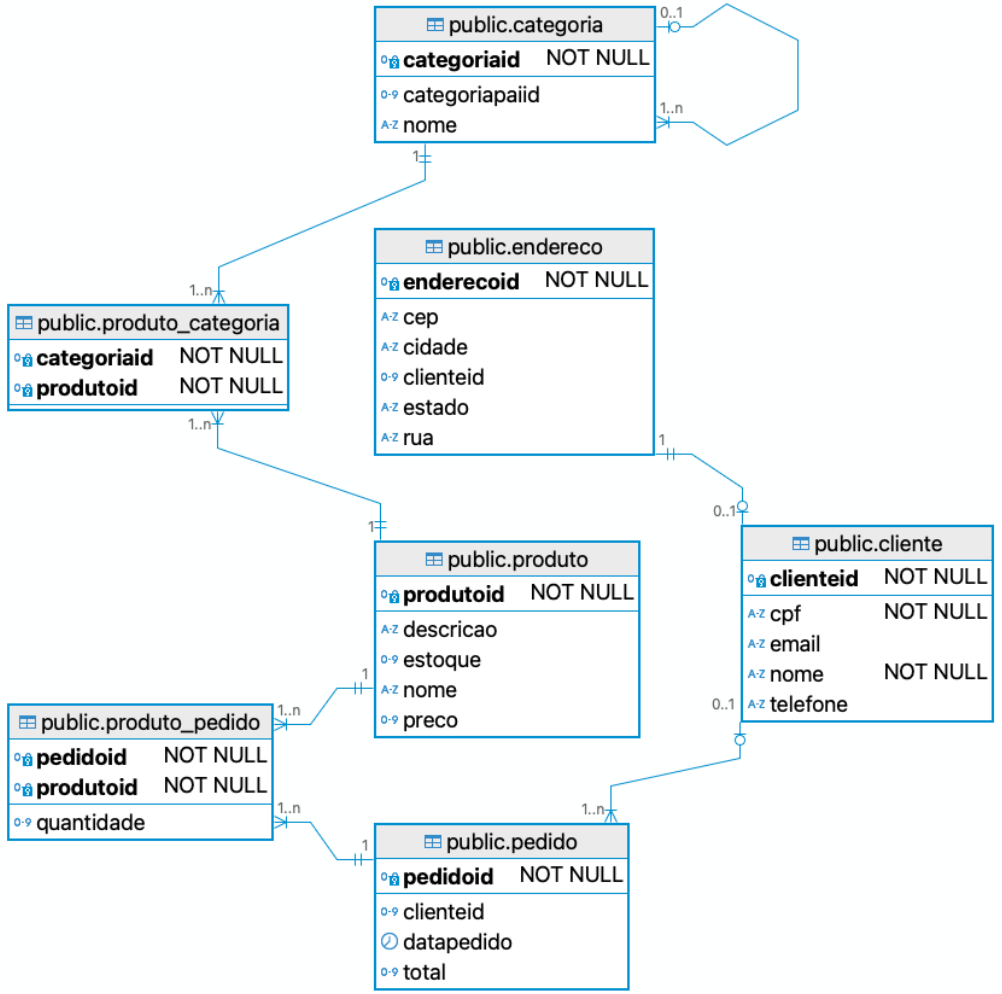
```
--Crie um banco de dados ecommerce;
CREATE DATABASE ecommerce;

--Altere seu nome para ecommercedb;
ALTER DATABASE ecommerce RENAME TO ecommercedb;

--Crie uma tabela dentro do schema public;
CREATE TABLE Cliente (
        ClienteID INTEGER PRIMARY KEY,
        Nome VARCHAR(100),
        Email VARCHAR(100),
        cpf CHAR(11),
        Telefone VARCHAR(15)
);

-- Apague o banco de dados ecommercedb
```

Notação de relacionamento entre tabelas

Notação pé de galinha (também conhecida como crow's foot)

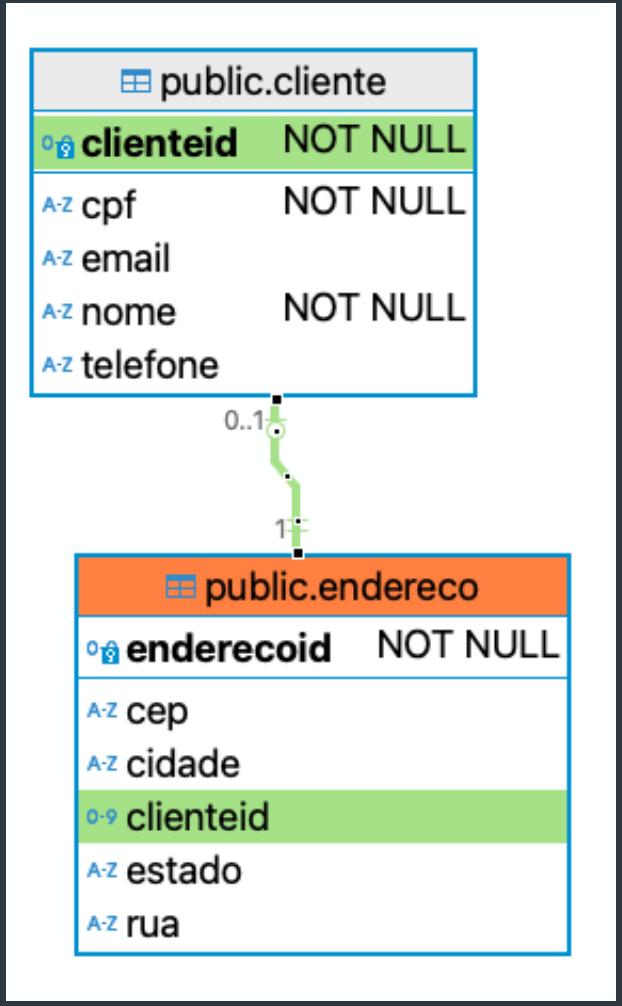






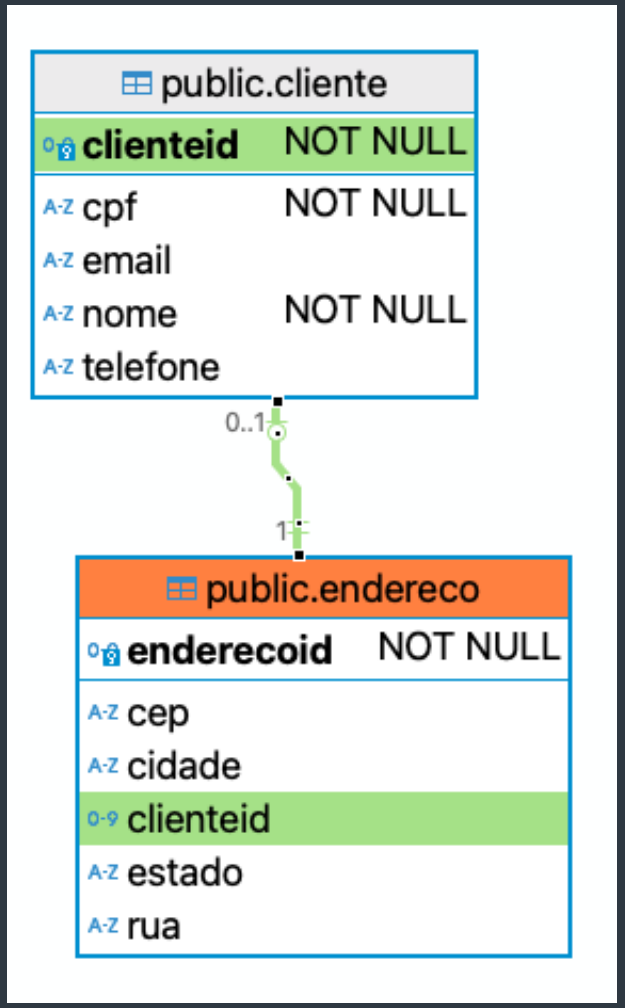
INTEGRIDADE REFERENCIAL
Relacionamento 1:1

```
-- Como fazer para o CPF virar uma chave candidata e não permitir valores nulos?
CREATE TABLE Cliente (
        ClienteID INTEGER PRIMARY KEY,
        Nome VARCHAR(100),
        Email VARCHAR(100),
        cpf CHAR(11),
        Telefone VARCHAR(15)
);

-- Como fazer incluir a chave estrangeira?
-- Como fazer para garantir um relacionamento 1:1?
CREATE TABLE Endereco (
        EnderecoID INTEGER PRIMARY KEY,
        ClienteID INTEGER,
        Rua VARCHAR(100),
        Cidade VARCHAR(50),
        Estado VARCHAR(50),
        CEP VARCHAR(10)
);
```



Relacionamento 1:1

```sql
-- Como fazer para o CPF virar uma chave candidata e não permitir valores nulos?
CREATE TABLE Cliente (
        ClienteID INTEGER PRIMARY KEY,
        Nome VARCHAR(100),
        Email VARCHAR(100),
        cpf CHAR(11),
        Telefone VARCHAR(15)
);

-- Como fazer incluir a chave estrangeira?
-- Como fazer para garantir um relacionamento 1:1?
CREATE TABLE Endereco (
        EnderecoID INTEGER PRIMARY KEY,
        ClienteID INTEGER,
        Rua VARCHAR(100),
        Cidade VARCHAR(50),
        Estado VARCHAR(50),
        CEP VARCHAR(10)
);

ALTER TABLE Endereco
ADD CONSTRAINT fk_cliente_endereco FOREIGN KEY (ClienteID)
REFERENCES Cliente(ClienteID);
```



Inserir valores nas tabelas Cliente e Endereço

```
INSERT INTO Cliente (ClienteID, Nome, Email, Cpf, Telefone) VALUES
(1,'João Silva', 'joao.silva@email.com','11111111111' , '123456789'),
(2,'Maria Oliveira', 'maria.oliveira@email.com', '22222222222', '987654321'),
(3,'Fernando', 'fernando@email.com', '33333333333', '987654321'),
(4,'Renata', 'renata.carlota@email.com', '44444444444', '123754323');
```

CLIENTE

clienteid	nome	email	cpf	telefone
1	João Silva	joao.silva@email.com	11111111111	123456789
2	Maria Oliveira	maria.oliveira@email.com	22222222222	987654321
3	Fernando	fernando@email.com	33333333333	987654321
4	Renata	renata.carlota@email.com	44444444444	123754323

```
INSERT INTO Endereco (EnderecoID, ClienteID, Rua, Cidade, Estado, CEP) VALUES
(1, 1, 'Rua A', 'Cidade X', 'Estado Y', '12345-678'),
(2, 2, 'Rua B', 'Cidade W', 'Estado Z', '98765-432'),
(3, 3, 'Rua C', 'Cidade C', 'Estado C', '98765-432');
```

ENDERECO

enderecoid	clienteid	rua	cidade	estado	cep
1	1	Rua A	Cidade X	Estado Y	12345-678
2	2	Rua B	Cidade W	Estado Z	98765-432
3	3	Rua C	Cidade C	Estado C	98765-432

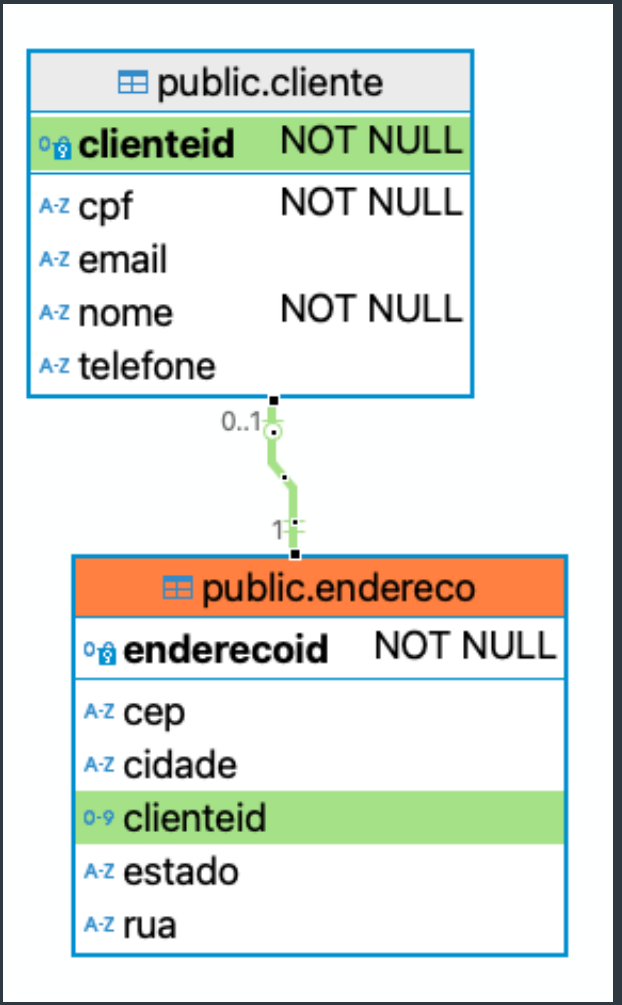
Como apagar uma tupla que faz referência a outra tabela?

CLIENTE

clienteid	nome	email	cpf	telefone
~~1~~	~~João Silva~~	~~joao.silva@email.com~~	~~11111111111~~	~~123456789~~
2	Maria Oliveira	maria.oliveira@email.com	22222222222	987654321
3	Fernando	fernando@email.com	33333333333	987654321
4	Renata	renata.carlota@email.com	44444444444	123754323

ENDERECO

enderecoid	clienteid	rua	cidade	estado	cep
1	1	Rua A	Cidade X	Estado Y	12345-678
2	2	Rua B	Cidade W	Estado Z	98765-432
3	3	Rua C	Cidade C	Estado C	98765-432



Restrição de integridade referencial
CREATE TABLE ... PRIMARY KEY ... FOREIGN KEY

- Comando que cria uma tabela considerando suas restrições de chave estrangeira
- A integridade referencial é especificada por meio da cláusula **FOREIGN KEY**

```
CREATE TABLE [<nome do esquema>.]<nome da tabela>...
FOREIGN KEY (<coluna1> [,...,<colunan>])
REFERENCES <tabela_ref>([<coluna_ref>[,...,<coluna_ref]])
[ON DELETE <acao_ref>] [ON UPDATE <acao_ref>];
```

<acao_ref>

- NO ACTION ou RESTRICTED → **impede** a ação na tabela mestre <tabela_ref>
- CASCADE → **propaga** a ação da tabela mestre
- SET NULL → valores de referencias alterados para nulo
- SET DEFAULT → valores de referencias alterados para default

Como apagar uma tupla que faz referência a outra tabela?

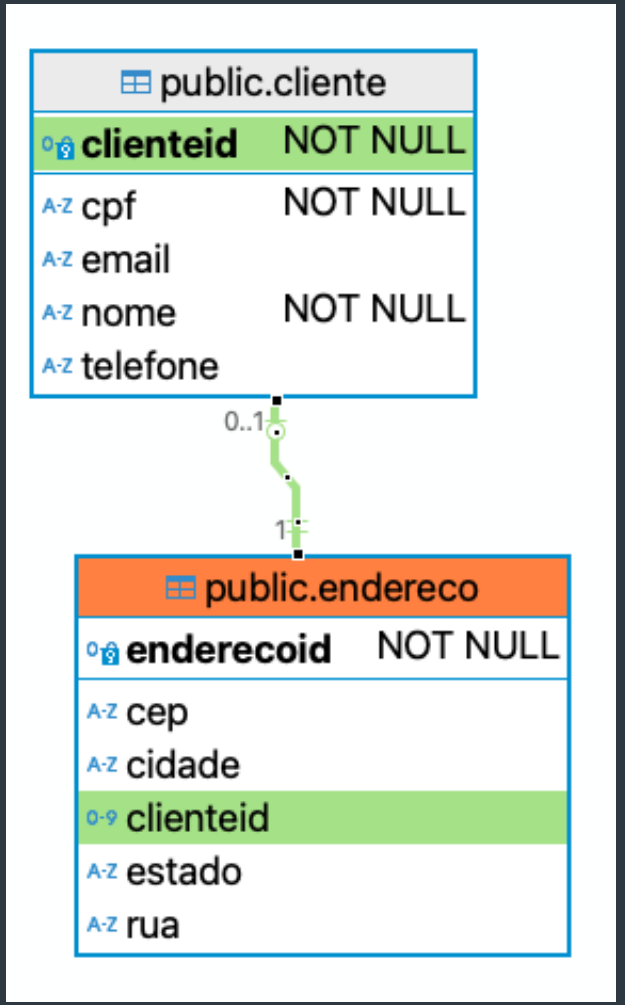
CLIENTE

clienteid	nome	email	cpf	telefone
~~1~~	~~João Silva~~	~~joao.silva@email.com~~	~~11111111111~~	~~123456789~~
2	Maria Oliveira	maria.oliveira@email.com	22222222222	987654321
3	Fernando	fernando@email.com	33333333333	987654321
4	Renata	renata.carlota@email.com	44444444444	123754323

ENDERECO

enderecoid	clienteid	rua	cidade	estado	cep
1	1	Rua A	Cidade X	Estado Y	12345-678
2	2	Rua B	Cidade W	Estado Z	98765-432
3	3	Rua C	Cidade C	Estado C	98765-432

```
DELETE FROM CLIENTE WHERE clienteid=1;
```



Como apagar uma tupla que faz referência a outra tabela? NO ACTION

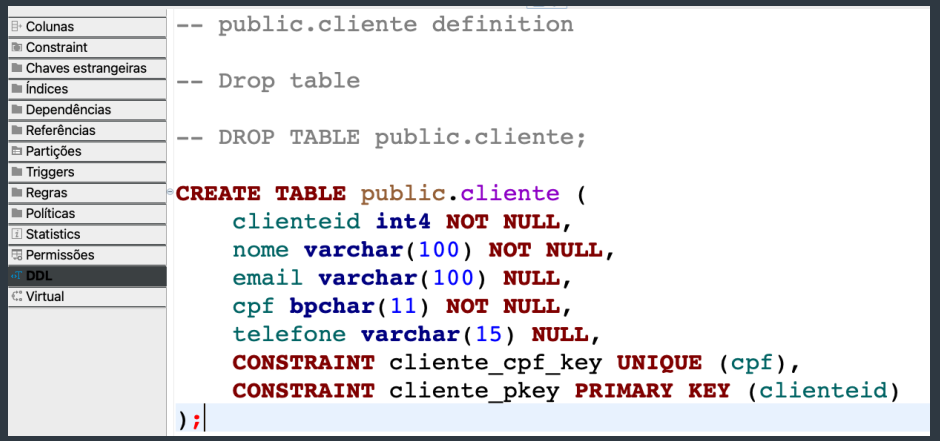
NO ACTION ou RESTRICTED → **impede** a ação na tabela mestre <tabela_ref>

CLIENTE

clienteid	nome	email	cpf	telefone
1	João Silva	joao.silva@email.com	11111111111	123456789
2	Maria Oliveira	maria.oliveira@email.com	22222222222	987654321
3	Fernando	fernando@email.com	33333333333	987654321
4	Renata	renata.carlota@email.com	44444444444	123754323

ENDERECO

enderecoid	clienteid	rua	cidade	estado	cep
1	1	Rua A	Cidade X	Estado Y	12345-678
2	2	Rua B	Cidade W	Estado Z	98765-432
3	3	Rua C	Cidade C	Estado C	98765-432



DELETE FROM CLIENTE WHERE clienteid=1;

Erro SQL [23503]: ERROR: update or delete on table "cliente" violates foreign key constraint "fk_cliente_endereco" on table "endereco"
  Detalhe: Key (clienteid)=(1) is still referenced from table "endereco".

Como apagar uma tupla que faz referência a outra tabela? CASCADE

CASCADE → **propaga** a ação da tabela mestre

```
----- CASCADE ----
ALTER TABLE endereco
DROP CONSTRAINT fk_cliente_endereco;

ALTER TABLE endereco
ADD CONSTRAINT fk_cliente_endereco
FOREIGN KEY (ClienteID) REFERENCES Cliente(ClienteID)
ON DELETE CASCADE;
```

```
CREATE TABLE public.endereco (
    enderecoid int4 NOT NULL,
    clienteid int4 NULL,
    rua varchar(100) NULL,
    cidade varchar(50) NULL,
    estado varchar(50) NULL,
    cep varchar(10) NULL,
    CONSTRAINT endereco_clienteid_key UNIQUE (clienteid),
    CONSTRAINT endereco_pkey PRIMARY KEY (enderecoid),
    CONSTRAINT fk_cliente_endereco FOREIGN KEY (clienteid)
    REFERENCES public.cliente(clienteid) ON DELETE CASCADE
);
```

```
DELETE FROM CLIENTE WHERE clienteid=1;
```

clienteid	nome	email	cpf	telefone
~~1~~	~~João Silva~~	~~joao.silva@email.com~~	~~11111111111~~	~~123456789~~
2	Maria Oliveira	maria.oliveira@email.com	22222222222	987654321
3	Fernando	fernando@email.com	33333333333	987654321
4	Renata	renata.carlota@email.com	44444444444	123754323

enderecoid	clienteid	rua	cidade	estado	cep
~~1~~	~~1~~	~~Rua A~~	~~Cidade X~~	~~Estado Y~~	~~12345-678~~
2	2	Rua B	Cidade W	Estado Z	98765-432
3	3	Rua C	Cidade C	Estado C	98765-432

Como apagar uma tupla que faz referência a outra tabela? SET NULL

SET NULL → valores de referências alterados para nulo

```
----- SET NULL ----
ALTER TABLE endereco
DROP CONSTRAINT fk_cliente_endereco;

ALTER TABLE endereco
ADD CONSTRAINT fk_cliente_endereco
FOREIGN KEY (ClienteID) REFERENCES Cliente(ClienteID)
ON DELETE SET NULL;
```

```
CREATE TABLE public.endereco (
    enderecoid int4 NOT NULL,
    clienteid int4 NULL,
    rua varchar(100) NULL,
    cidade varchar(50) NULL,
    estado varchar(50) NULL,
    cep varchar(10) NULL,
    CONSTRAINT endereco_clienteid_key UNIQUE (clienteid),
    CONSTRAINT endereco_pkey PRIMARY KEY (enderecoid),
    CONSTRAINT fk_cliente_endereco FOREIGN KEY (clienteid)
    REFERENCES public.cliente(clienteid) ON DELETE SET NULL
);
```

DELETE FROM CLIENTE WHERE clienteid=2;

clienteid	nome	email	cpf	telefone
~~2~~	~~Maria Oliveira~~	~~maria.oliveira@email.com~~	~~22222222222~~	~~987654321~~
3	Fernando	fernando@email.com	33333333333	987654321
4	Renata	renata.carlota@email.com	44444444444	123754323

enderecoid	clienteid	rua	cidade	estado	cep
2	NULL	Rua B	Cidade W	Estado Z	98765-432
3	3	Rua C	Cidade C	Estado C	98765-432

Como apagar uma tupla que faz referência a outra tabela? SET DEFAULT

SET DEFAULT → valores de referencias alterados para default

```
----- SET DEFAULT ----
ALTER TABLE endereco
DROP CONSTRAINT fk_cliente_endereco;

ALTER TABLE endereco
ALTER COLUMN clienteid SET DEFAULT 0;

ALTER TABLE endereco
ADD CONSTRAINT fk_cliente_endereco
FOREIGN KEY (ClienteID) REFERENCES Cliente(ClienteID)
ON DELETE SET DEFAULT;
```

```
CREATE TABLE public.endereco (
    enderecoid int4 NOT NULL,
    clienteid int4 DEFAULT '-1'::integer NOT NULL,
    rua varchar(100) NULL,
    cidade varchar(50) NULL,
    estado varchar(50) NULL,
    cep varchar(10) NULL,
    CONSTRAINT clienteid_ak UNIQUE (clienteid),
    CONSTRAINT endereco_pkey PRIMARY KEY (enderecoid),
    CONSTRAINT fk_cliente_endereco FOREIGN KEY (clienteid)
    REFERENCES public.cliente(clienteid) ON DELETE SET DEFAULT
);
```

DELETE FROM CLIENTE WHERE clienteid=3;

clienteid	nome	email	cpf	telefone
~~3~~	~~Fernando~~	~~fernando@email.com~~	~~33333333333~~	~~987654321~~
4	Renata	renata.carlota@email.com	44444444444	123754323

enderecoid	clienteid	rua	cidade	estado	cep
2	NULL	Rua B	Cidade W	Estado Z	98765-432
3	3	Rua C	Cidade C	Estado C	98765-432

Como apagar uma tupla que faz referência a outra tabela? SET DEFAULT

SET DEFAULT → valores de referencias alterados para default

```
----- SET DEFAULT ----
ALTER TABLE endereco
DROP CONSTRAINT fk_cliente_endereco;

ALTER TABLE endereco
ALTER COLUMN clienteid SET DEFAULT 0;

ALTER TABLE endereco
ADD CONSTRAINT fk_cliente_endereco
FOREIGN KEY (ClienteID) REFERENCES Cliente(ClienteID)
ON DELETE SET DEFAULT;
```

NULL

```
CREATE TABLE public.endereco (
    enderecoid int4 NOT NULL,
    clienteid int4 DEFAULT '-1'::integer NOT NULL,
    rua varchar(100) NULL,
    cidade varchar(50) NULL,
    estado varchar(50) NULL,
    cep varchar(10) NULL,
    CONSTRAINT clienteid_ak UNIQUE (clienteid),
    CONSTRAINT endereco_pkey PRIMARY KEY (enderecoid),
    CONSTRAINT fk_cliente_endereco FOREIGN KEY (clienteid)
    REFERENCES public.cliente(clienteid) ON DELETE SET DEFAULT
);
```

```
DELETE FROM CLIENTE WHERE clienteid=3;
```

clienteid	nome	email	cpf	telefone
~~3~~	~~Fernando~~	~~fernando@email.com~~	~~33333333333~~	~~987654321~~
4	Renata	renata.carlota@email.com	44444444444	123754323

enderecoid	clienteid	rua	cidade	estado	cep
2	NULL	Rua B	Cidade W	Estado Z	98765-432
3	NULL	Rua C	Cidade C	Estado C	98765-432

Restrição de CHECK

- A restrição de check (verificação) especifica uma condição que precisa ser satisfeita por cada tupla da relação.
- O predicado de uma cláusula CKECK pode ser uma subconsulta SQL

```
CREATE TABLE Produto (
        ProdutoID INTEGER PRIMARY KEY,
        Nome varchar(255) NOT NULL,
        Descricao TEXT,
        Preco numeric(10, 2) NOT NULL CHECK (Preco > 0),
        Estoque integer NOT NULL CHECK (Estoque >= 0 AND Estoque <= 1000)
);

-- Produto
INSERT INTO Produto (ProdutoID, Nome, Descricao, Preco, Estoque) VALUES
(1,'Produto 1', 'Descrição do Produto 1', 100.00, 50),
(2, 'Produto 2', 'Descrição do Produto 2', 200.00, 30);

-- ERRO
INSERT INTO Produto (ProdutoID, Nome, Descricao, Preco, Estoque) VALUES
(3,'Produto 3', 'Descrição do Produto 3', 100.00, 1001);
```

```
Erro SQL [23514]: ERROR: new row for relation "produto" violates check constraint
"produto_estoque_check"
  Detalhe: Failing row contains (3, Produto 3, Descrição do Produto 3, 100.00, 1001).
```

Definição de domínio
[CREATE | DROP] DOMAIN

Cria um domínio para um tipo de dado.

- Sintaxe:

```
[CREATE | DROP] DOMAIN <nome_do_domínio> AS <tipo_do_domínio> <expressão default> <restrições>;
```

- **Vantagens de Usar Domínios Específicos**
  - **Reutilização:** Uma vez criado, o domínio pode ser reutilizado em várias tabelas, garantindo consistência sem precisar redefinir as mesmas restrições repetidamente.
  - **Centralização de Regras:** Facilita a manutenção, pois as regras de validação de dados são centralizadas no domínio.
  - **Simplificação:** Reduz a complexidade dos scripts de criação de tabelas, tornando-os mais legíveis e concisos.

```
CREATE DOMAIN preco_positivo AS numeric(10, 2) CHECK (VALUE > 0);

CREATE TABLE Produto (
        ProdutoID INTEGER PRIMARY KEY,
        Nome varchar(255) NOT NULL,
        Preco preco_positivo NOT NULL, -- Usando o domínio específico
        Estoque integer NOT NULL CHECK (Estoque >= 0)
);
```